
TP 2.1 - GENERADORES PSEUDOALEATORIOS

Juan I. Torres
Cátedra Simulación
UTN - FRRO
Zeballos 1341, S2000
orkuan@gmail.com

28 de Abril, 2020

ABSTRACT

El siguiente documento tiene por objetivo detallar el trabajo de clase que debe realizarse para introducirnos en uno de los elementos fundamentales para gran parte de las simulaciones, esto son los **generadores de números pseudoaleatorios**.

Keywords Simulación · Trabajo práctico · generadores pseudoaleatorios

1 Introducción

Un número pseudoaleatorio es un número generado en un proceso que parece producir números al azar, pero no lo hace realmente. Las secuencias de números pseudo-aleatorios no muestran ningún patrón o regularidad aparente desde un punto de vista estadístico, a pesar de haber sido generadas por un algoritmo completamente determinista, en el que las mismas condiciones iniciales producen siempre el mismo resultado.

2 Descripción del trabajo

El trabajo consiste en construir programas en lenguaje Python 3.x que generen números pseudoaleatorios y que estos se comporten como se espera. Para esto se debe tener en cuenta lo siguientes temas a investigar:

- Generadores de números aleatorios reales.
- Generadores de números pseudoaleatorios (Método de los cuadrados, GCLs, otros).
- Test para determinar el comportamiento de los generadores.

Se pide programar al menos dos generadores de números pseudoaleatorios en particular el generador GCL del cual se debe testear con al menos cuatro pruebas para determinar la calidad de generación. También se pide comparar los generadores programados con otros, incluyendo el que posee el lenguaje Python.

A modo de introducción y ejemplos de código se deja un link en la sección de **Recursos online obligatorios**.

2.1 Exposición de los resultados y análisis de los mismos

Los resultados se deben analizar y luego concluir sobre su comportamiento obtenido y esperado.

Se aconseja el uso de tablas para mostrar los datos de los distintos generadores y pruebas. En otros casos puede ser más convenientes el uso de imágenes. Ver los siguientes ejemplos:

Generador	Tipo de Generador	Test X	Test Y	...
GCL	Pseudoaleatorio	0.4	OK	...
ERNIE	Aleatorio	0.2	ERROR	...
...

Tabla 1: Ejemplo de exposición de resultados

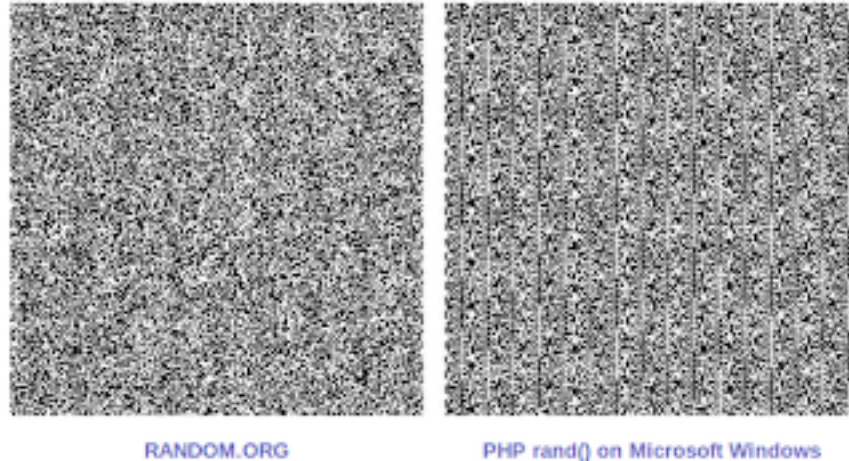


Figura 1: Comparación entre dos generadores. Imagen tomada de <http://www.random.org/analysis/>.

2.2 Presentación del trabajo y entrega

Este y los siguientes TPs se presenta obligatoriamente en formato \LaTeX . Una manera cómoda de trabajar es mediante un IDE (entorno de desarrollo integrado) el cual puede ser online o local. La ventaja de trabajar online es la posibilidad de que el resto del grupo aporte y corrija directamente y simultáneamente.

Una de las plataformas online más conocidas es Overleaf. Por el otro lado, en forma local tenemos, para los que trabajan con Linux distribución Ubuntu el muy conocido Texmaker. En Windows debe instalarse primero el compilador Miktex, y posteriormente puede instalarse Texmaker o TeXStudio. VSCode también tiene utilidades para escribir en LaTeX.

El contenido mínimo a entregar es:

- Código completo en Python 3.x.
- Informe en formato Latex con introducción, gráficas, fórmulas empleadas, conclusiones, referencias (hay un apartado para esto) y cualquier otra información que se quiera agregar.

La fecha de entrega es el día **16/05/2020**.

3 Recursos online obligatorios

Volveremos a usar el template \LaTeX de la Cornell University por su sencillez:

<https://es.overleaf.com/latex/templates/style-and-template-for-preprints-arxiv-bio-arxiv/pkzcrhzcxdxmc>

Un resumen que va a ayudar a encontrar mucha otra información:

<https://tereom.github.io/est-computacional-2018/numeros-pseudoaleatorios.html>

Acá encontrarán información de muchos tests:

<https://www.random.org/analysis/>

En particular:

<https://www.random.org/analysis/Analysis2005.pdf>

4 Recursos online sugeridos

<https://www.youtube.com/watch?v=FtEe0IlK6Hc>